IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Mang'oli Martin
19 September 2022

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- This project involved coming up with a data science model that predicts whether a rocket will land successfully during the first launch. Successful first stage landing enables SpaceX to reuse it during the subsequent launches hence reducing the cost of rocket launches by more than 100 million dollars.
- We collected data of the previous launches from the SpaceX website through their APIs to enable us predict the outcome of the launch.
- After cleaning the data we subjected it to four machine learning algorithms: Logistic Regression, Support Vector Machines, Decision Tree, and K Nearest Neighbors.
- All the algorithms gave higher accuracy except Decision Tree. Therefore the company could rely on either of the three models to predict whether the first launch will be successful or not.

Introduction

- Space X advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because Space X can reuse the first stage.
- Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against space X for a rocket launch.
- In this project, we created a machine learning pipeline to predict if the first stage will land given the data from the previous launches.

Section 1

Methodology

Methodology

Executive Summary

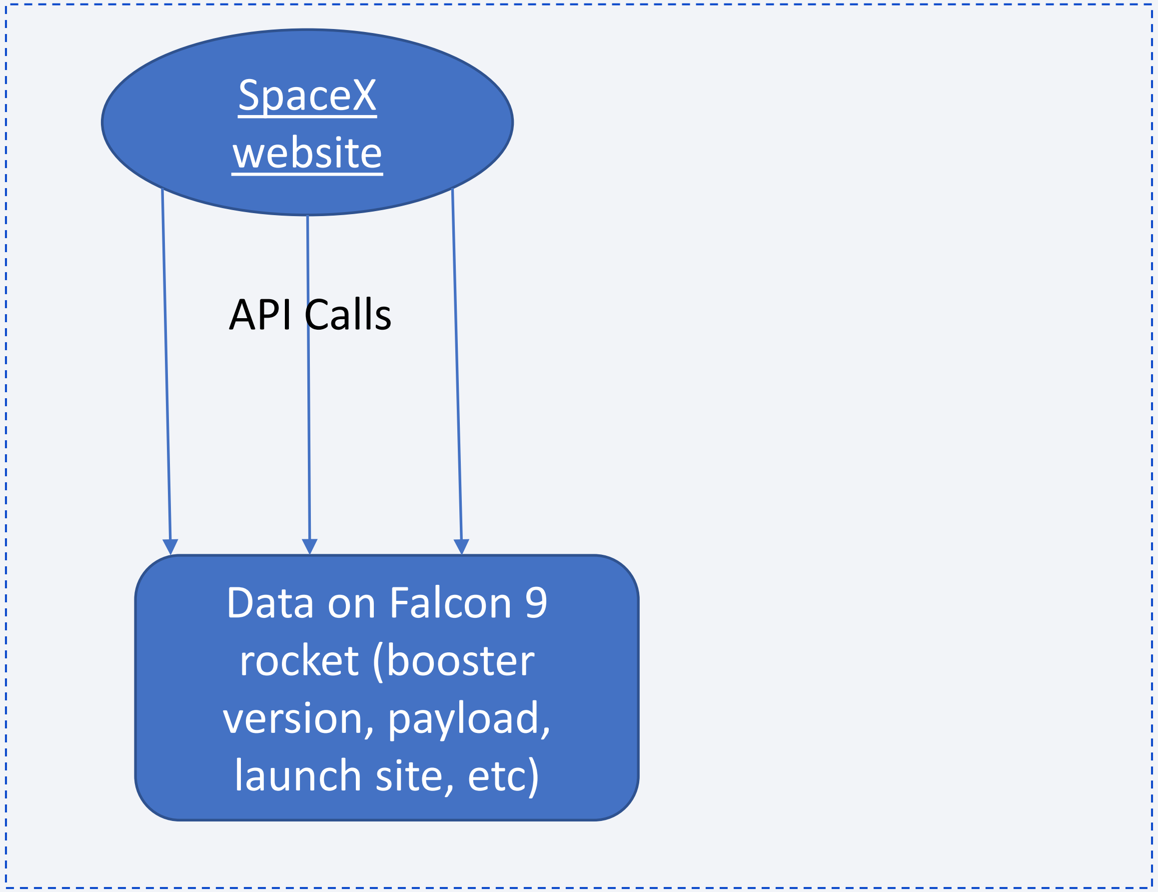
- Data collection methodology:
  - We collected data on the characteristics of the Falcon 9 rockets from SpaceX website using their APIS.
  - We also collected Falcon 9 historical launch records from a Wikipedia page by performing web scraping.
- Perform data wrangling
  - We replaced missing values in the Payload column with the average mass of the payload.
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models

Data Collection

- One set of data about characteristics of Falcon 9 rockets was collected from SpaceX website using their own APIs.
- Data on historical launches whether they were successful or not was collected from Wikipedia through web scrapping.

Data Collection – SpaceX API

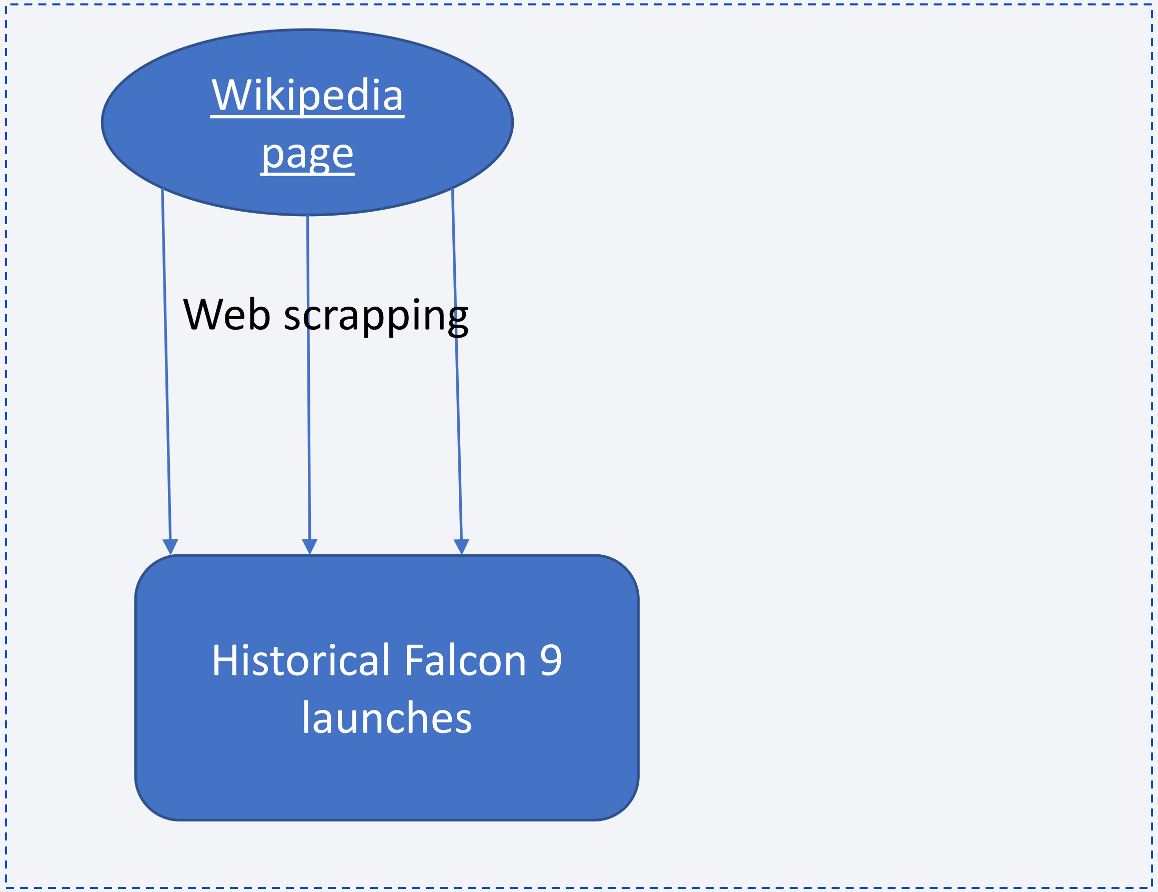
- SpaceX website has several API end points that we accessed to get the needed information:
  - Rockets API gives the booster name
  - Launchpads API gives the site being used, the longitude, and the latitude
  - Payloads API gives the mass of the payload and the orbit that it is going to
  - Cores API gives the core data about the rocket
  - Past API gives the past launch data
- The GitHub URL of the completed SpaceX API calls notebook (https://github.com/mangsmato/IBM-Applied-Data-Science-Capstone/blob/master/Data%20Collection%20API.ipynb)



Data Collection - Scraping

- Using requests and BeatifulSoup libraries we performed web scraping to collect Falcon 9 historical launch records from a Wikipedia page titled [List of Falcon 9 and Falcon Heavy launches]

- The GitHub URL of the completed web scraping notebook ([https://github.com/mangsmato/IBM-Applied-Data-Science-Capstone/blob/master/Data%20Collection%20with%20Web%20Scraping.ipynb])



Data Wrangling

- Once we got the data from the SpaceX API calls we filtered it to contain only Falcon 9 launches.
- We left the LandingPad column to retain missing values to represent when landing pads were not used.
- We replaced missing values on the payload data with the average payload mass.
- The GitHub URL of the data wrangling on SpaceX API calls notebook (https://github.com/mangsmato/IBM-Applied-Data-Science-Capstone/blob/master/Data%20Collection%20API.ipynb)

EDA with Data Visualization

- Using a scatter plot we noted that as the flight number increases, the first stage is more likely to land successfully.
- The more massive the payload, the less likely the first stage will return.
- It also showed VAFB SLC 4E launch site had the best success rate with a few flights. The success rate at CCAFS LC-40 launch site is low compared to the other two launch sites (KSC LC-39A and VAFB SLC 4E) hence there's need to increase the number of lights for the launch to be successful.
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS orbits.
- A bar chart enabled us to know that ES-L1, GEO, HEO, and SSO orbits had high success rates.
- A line graph showed an increase in the success rate since 2013 till 2020
- The GitHub URL of the ompleted EDA with data visualization notebook (https://github.com/mangsmato/IBM-Applied-Data-Science-Capstone/blob/master/EDA%20with%20Data%20Visualization.ipynb)

EDA with SQL

- select * from cnm23816.spacexdataset to extract the entire dataset
- select distinct launch_site from cnm23816.spacexdataset to display the names of the unique launch sites in the space mission.
- select * from cnm23816.spacexdataset where upper(launch_site) like 'CCA%' LIMIT 5 to display 5 records where launch sites begin with the string 'CCA'.
- select SUM(payload_mass__kg_) total_payload_mass from cnm23816.spacexdataset WHERE UPPER(customer) = 'NASA (CRS)' to display the total payload mass carried by boosters launched by NASA (CRS).
- select AVG(payload_mass__kg_) average_payload_mass from cnm23816.spacexdataset where upper(booster_version) like 'F9 V1.1' to display average payload mass carried by booster version F9 v1.1.
- select MIN(DATE) MIN_DATE from cnm23816.spacexdataset where landing__outcome = 'Success (ground pad)' to List the date when the first successful landing outcome in ground pad was achieved.
- select DISTINCT booster_version from cnm23816.spacexdataset where landing__outcome = 'Success (drone ship)' AND payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000 to List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- select mission_outcome, COUNT(*) outcomes from cnm23816.spacexdataset GROUP BY mission_outcome to list the total number of successful and failure mission outcomes
- select DISTINCT booster_version, payload_mass__kg_ from cnm23816.spacexdataset where payload_mass__kg_ = (select MAX(payload_mass__kg_) from cnm23816.spacexdataset) to List the names of the booster_versions which have carried the maximum payload mass.
- select DATE, landing__outcome, booster_version, launch_site from cnm23816.spacexdataset where landing__outcome = 'Failure (drone ship)' and year(DATE) = 2015 to List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015.
- select landing__outcome, COUNT(*) outcomes from cnm23816.spacexdataset where date between '2010-06-04' and '2017-03-20' GROUP BY landing__outcome ORDER BY COUNT(*) DESC to Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order
- The GitHub URL of the completed EDA with SQL notebook (https://github.com/mangsmato/IBM-Applied-Data-Science-Capstone/blob/master/EDA%20with%20SQL.ipynb)

Build an Interactive Map with Folium

- Added the following map objects to the folium map:
  - Added each site's location on a map using site's latitude and longitude coordinates.
  - Added a circle for each launch site in the data frame.
  - Created markers for all launch records. If a launch was successful (class=1), then we used a green marker and if a launch was failed, we used a red marker (class=0)
  - Drew lines between a launch site to its closest city, railway, highway, coastline to explore and analyze the proximities of launch sites.
- The GitHub URL of the completed interactive map with Folium map (https://github.com/mangsmato/IBM-Applied-Data-Science-Capstone/blob/master/Interactive%20Visual%20Analytics%20with%20Folium.ipynb)

Build a Dashboard with Plotly Dash

- Added the following plots/graphs and interactions to the dashboard:
  - Added a dropdown list to enable Launch Site selection. The default select value is for ALL sites.
  - Added a pie chart to show the total successful launches count for all sites. If a specific launch site was selected, show the Success vs. Failed counts for the site.
  - Added a slider to select payload range
  - Added a scatter chart to show the correlation between payload and launch success
- The GitHub URL of the completed Plotly Dash (https://github.com/mangsmato/IBM-Applied-Data-Science-Capstone/blob/master/spacex_dash_app.py)

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
  - Imported key libraries (pandas, numpy, scikit-learn, matplotlib) and defined auxiliary functions.
  - Loaded the data using pandas library.
  - Split the data into features and labels. Standardized the features.
  - Split the data into training and testing sets with 80% training and 20% testing.
  - Trained four classification models using the data: Logistic Regression, Support Vector Machines (SVM), Decision Tree, K Nearest Neighbors (KNN).
  - Used Grid Search CV to select best parameters for each model.
  - Using the confusion matrix picked the best model.
- The GitHub URL of the completed predictive analysis (https://github.com/mangsmato/IBM-Applied-Data-Science-Capstone/blob/master/SpaceX_Machine%20Learning%20Prediction.ipynb)

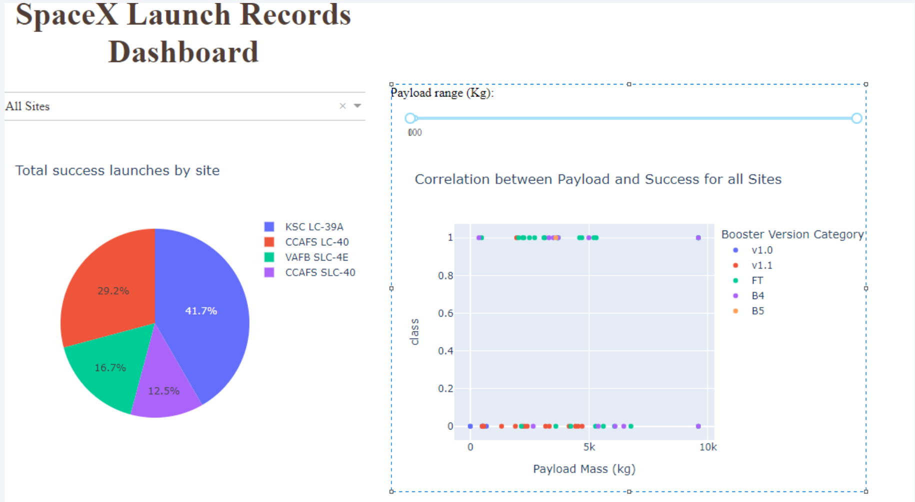
Results

- Exploratory data analysis results

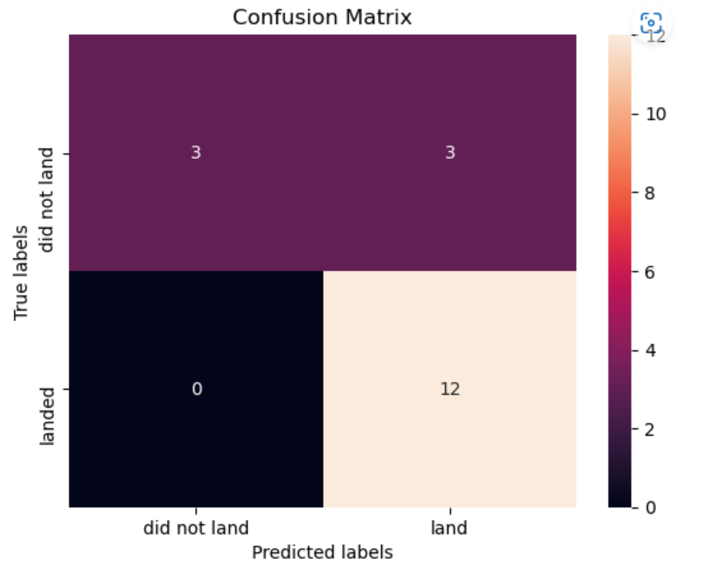
```
# Apply value_counts() on column LaunchSite
df.LaunchSite.value_counts().rename_axis('LaunchSite').reset_index(name='Launches')
```

	LaunchSite	Launches
0	CCAFS SLC 40	55
1	KSC LC 39A	22
2	VAFB SLC 4E	13

- Interactive analytics demo in screenshots



Predictive analysis results.

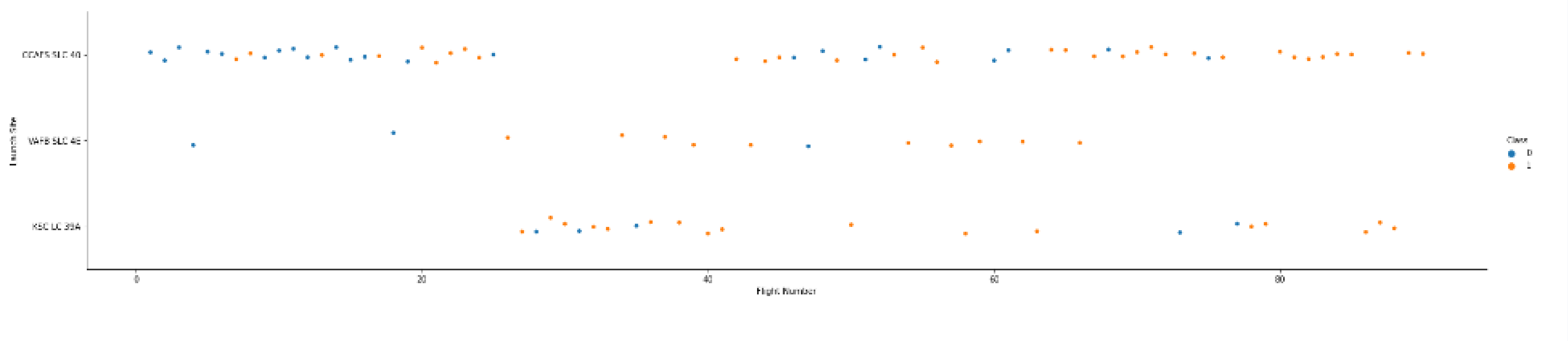


Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

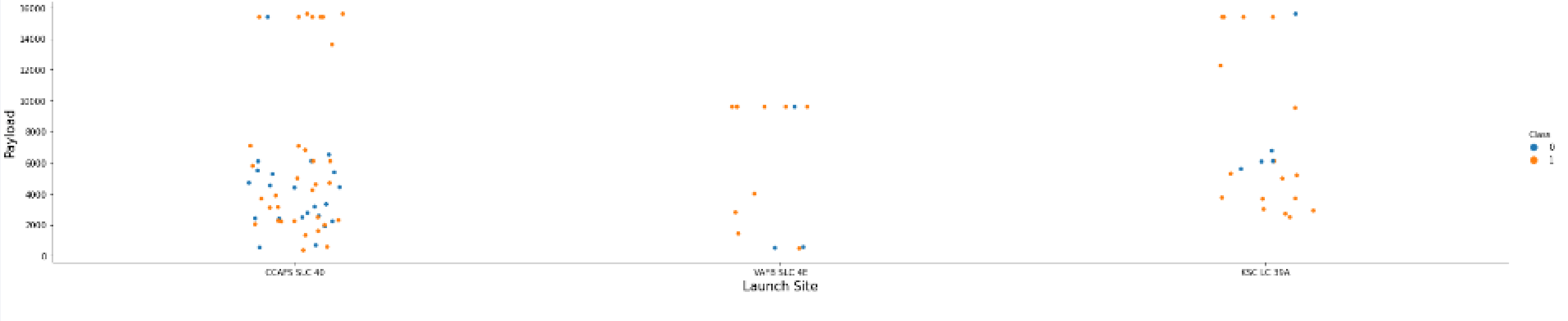
- A scatter plot of Flight Number vs. Launch Site



- The success rate at CCAFS LC-40 is low compared to the other two launch sites (KSC LC-39A and VAFB SLC 4E) hence there's need to increrase the number of lights for the launch to be successful
- VAFB SLC 4E has the best success rate with a few flights

Payload vs. Launch Site

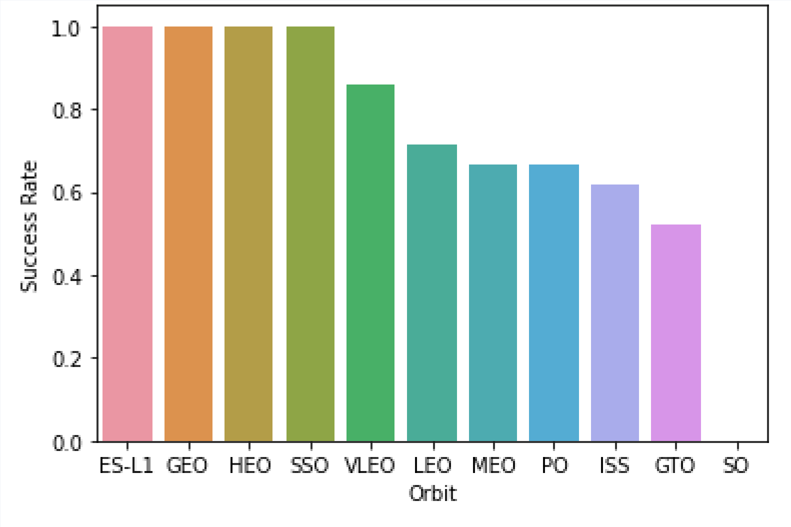
- A scatter plot of Payload vs. Launch Site



- For the VAFB-SLC launch site there are no rockets launched for heavy payload mass (greater than 10000).

Success Rate vs. Orbit Type

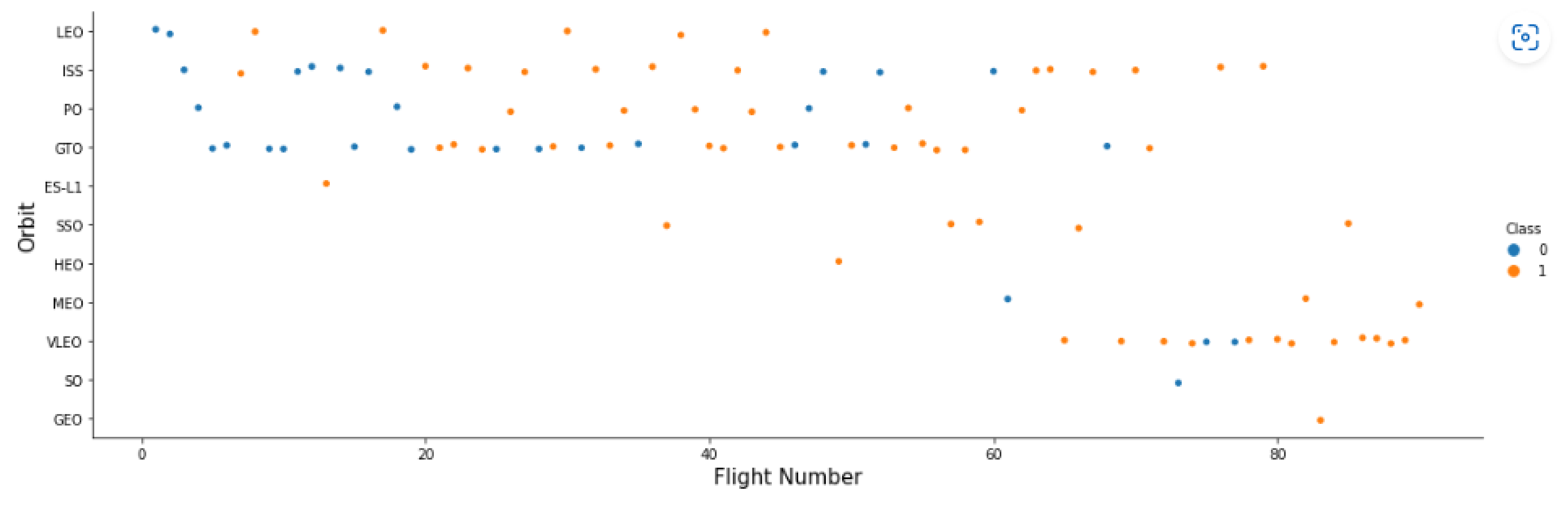
- Abar chart for the success rate of each orbit type



- The following orbits have a high success rate:
  - ES-L1
  - GEO
  - HEO
  - SSO

Flight Number vs. Orbit Type

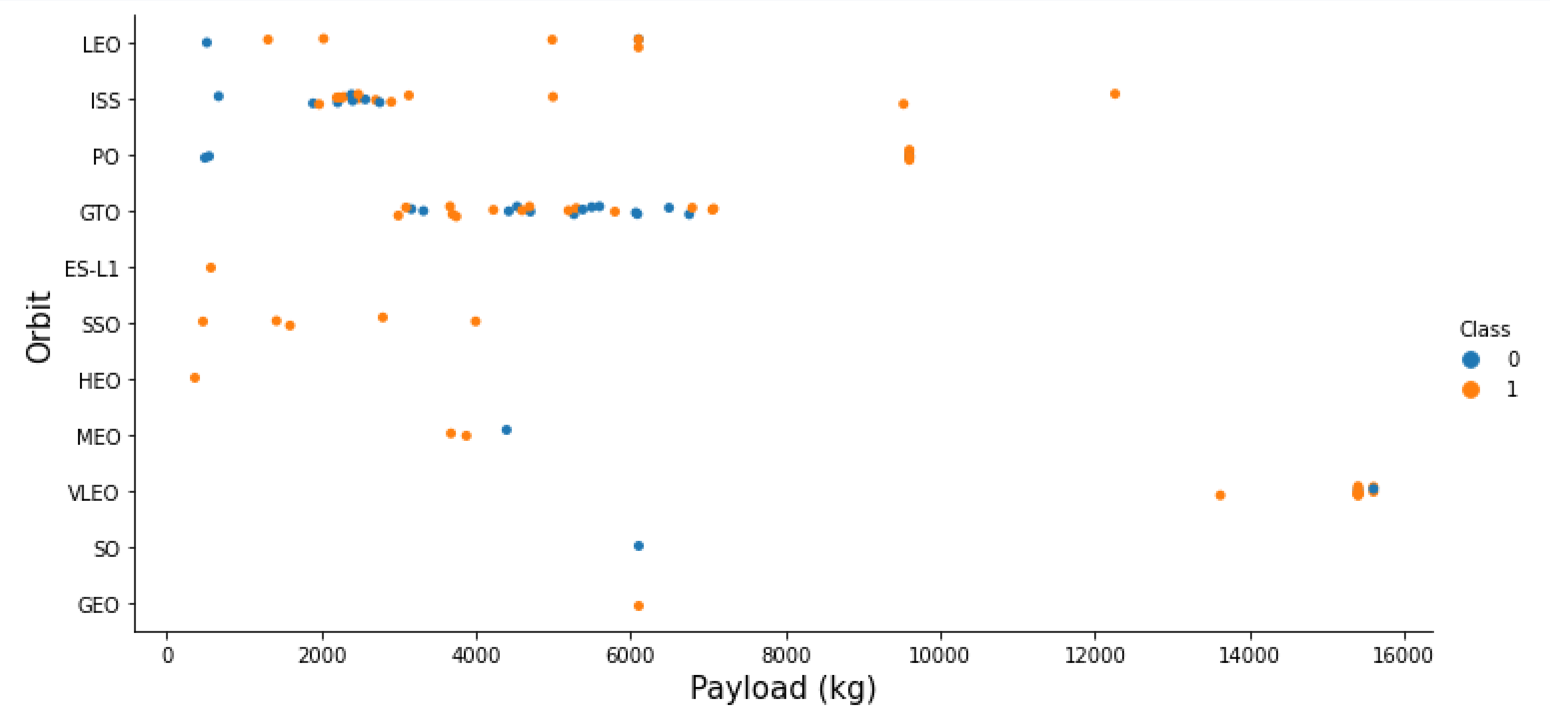
- A scatter point of Flight number vs. Orbit type



- In the LEO orbit the success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Payload vs. Orbit Type

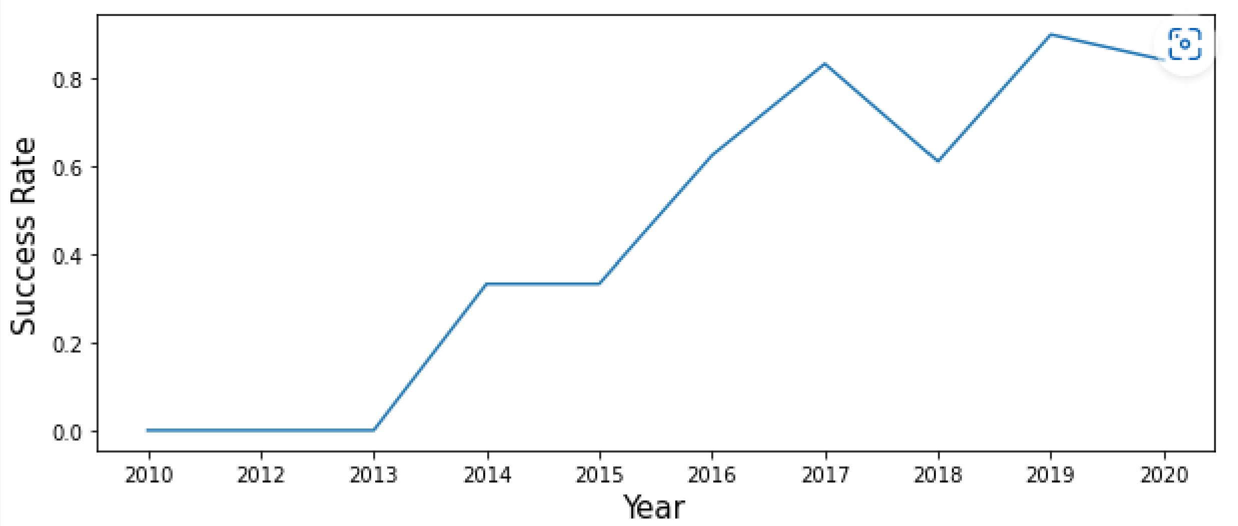
- A scatter point of payload vs. orbit type



- With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Launch Success Yearly Trend

- A line chart of yearly average success rate



- The success rate since 2013 kept increasing till 2020

All Launch Site Names

- The names of the unique launch sites:

```
%%sql
select distinct launch_site from cnm23816.spacexdataset
```

```
 * ibm_db_sa://cnm23816:***@8e359033-a1c9-4643-82ef-8ac06
Done.
```

launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- 5 records where launch sites begin with `CCA`

```
%%sql
select * from cnm23816.spacexdataset
where upper(launch_site) like 'CCA%'
LIMIT 5
```

```
 * ibm_db_sa://cnm23816:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:30120/bludb
Done.
```

]:

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

Total Payload Mass

- The total payload carried by boosters from NASA

```
%%sql
select SUM(payload_mass__kg_) total_payload_mass
from cnm23816.spacexdataset
WHERE UPPER(customer) = 'NASA (CRS)'
```

```
 * ibm_db_sa://cnm23816:***@8e359033-a1c9-4643-82ef
Done.
```

]:

total_payload_mass
45596

Average Payload Mass by F9 v1.1

- The average payload mass carried by booster version F9 v1.1

```
%%sql
select AVG(payload_mass__kg_) average_payload_mass
from cnm23816.spacexdataset
where upper(booster_version) like 'F9 V1.1'
```

```
 * ibm_db_sa://cnm23816:***@8e359033-a1c9-4643-82ef-8
Done.
```

average_payload_mass
2928

First Successful Ground Landing Date

- The dates of the first successful landing outcome on ground pad

```
%%sql
select MIN(DATE) MIN_DATE
from cnm23816.spacexdataset
where landing__outcome = 'Success (ground pad)'
```

```
 * ibm_db_sa://cnm23816:***@8e359033-a1c9-4643-82ef-
Done.
```

min_date
2015-12-22

Successful Drone Ship Landing with Payload between 4000 and 6000

- The names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

```
%%sql
select DISTINCT booster_version
from cnm23816.spacexdataset
where landing__outcome = 'Success (drone ship)'
AND payload_mass__kg_ > 4000 AND payload_mass__kg_ < 6000
```

```
 * ibm_db_sa://cnm23816:***@8e359033-a1c9-4643-82ef-8ac06f5
Done.
```

]:

booster_version
F9 FT B1021.2
F9 FT B1031.2
F9 FT B1022
F9 FT B1026

Total Number of Successful and Failure Mission Outcomes

- The total number of successful and failure mission outcomes

```
%%sql
select mission_outcome, COUNT(*) outcomes
from cnm23816.spacexdataset
GROUP BY mission_outcome
```

```
 * ibm_db_sa://cnm23816:***@8e359033-a1c9-4643-
Done.
```

]:

mission_outcome	outcomes
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Boosters Carried Maximum Payload

- The names of the booster which have carried the maximum payload mass.

```
%%sql
select DISTINCT booster_version, payload_mass__kg_
from cnm23816.spacexdataset
where payload_mass__kg_ = (
    select MAX(payload_mass__kg_)
    from cnm23816.spacexdataset
)
```

```
 * ibm_db_sa://cnm23816:***@8e359033-a1c9-4643-82ef-8
Done.
```

booster_version	payload_mass__kg_
F9 B5 B1048.4	15600
F9 B5 B1048.5	15600
F9 B5 B1049.4	15600
F9 B5 B1049.5	15600
F9 B5 B1049.7	15600
F9 B5 B1051.3	15600
F9 B5 B1051.4	15600
F9 B5 B1051.6	15600
F9 B5 B1056.4	15600
F9 B5 B1058.3	15600
F9 B5 B1060.2	15600
F9 B5 B1060.3	15600

2015 Launch Records

- The failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

```
%%sql
select DATE, landing__outcome, booster_version, launch_site
from cnm23816.spacexdataset
where landing__outcome = 'Failure (drone ship)'
and year(DATE) = 2015
```

```
 * ibm_db_sa://cnm23816:***@8e359033-a1c9-4643-82ef-8ac06f5107eb.
Done.
```

DATE	landing__outcome	booster_version	launch_site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

- The count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

```
%%sql
select landing__outcome, COUNT(*) outcomes
from cnm23816.spacexdataset
where date between '2010-06-04' and '2017-03-20'
GROUP BY landing__outcome
ORDER BY COUNT(*) DESC
```

```
 * ibm_db_sa://cnm23816:***@8e359033-a1c9-4643-82ef-8ac06
Done.
```

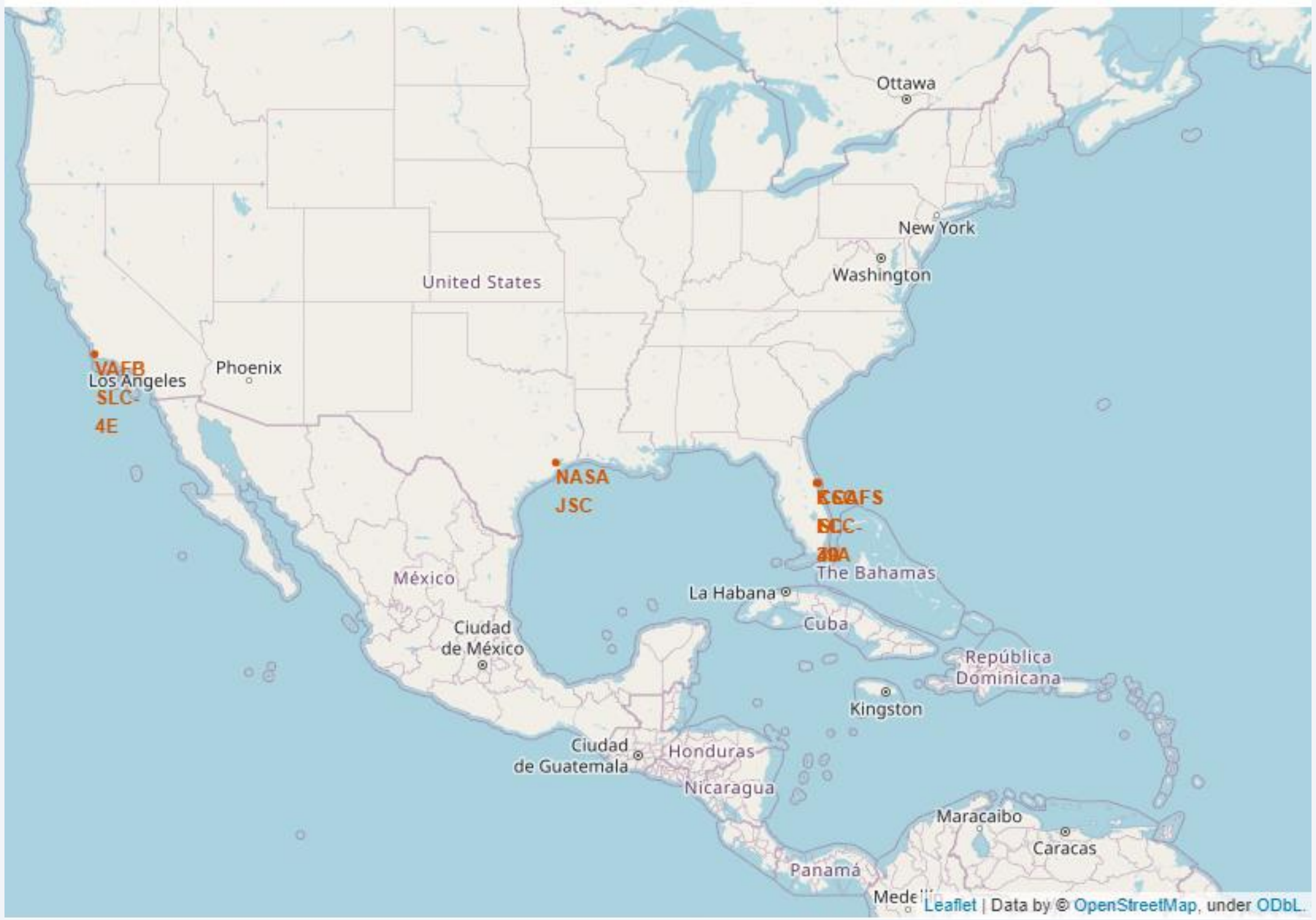
]:

landing__outcome	outcomes
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

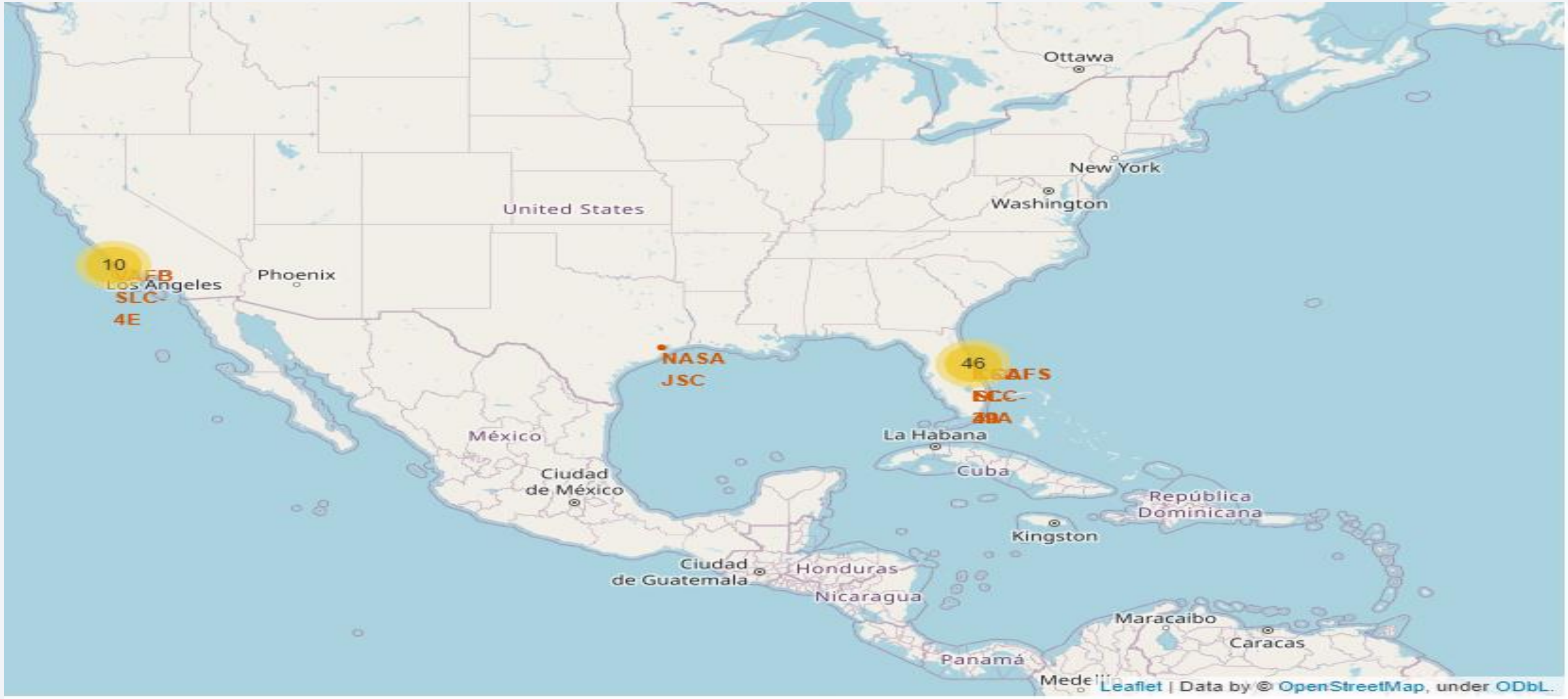
Section 3

Launch Sites Proximities Analysis

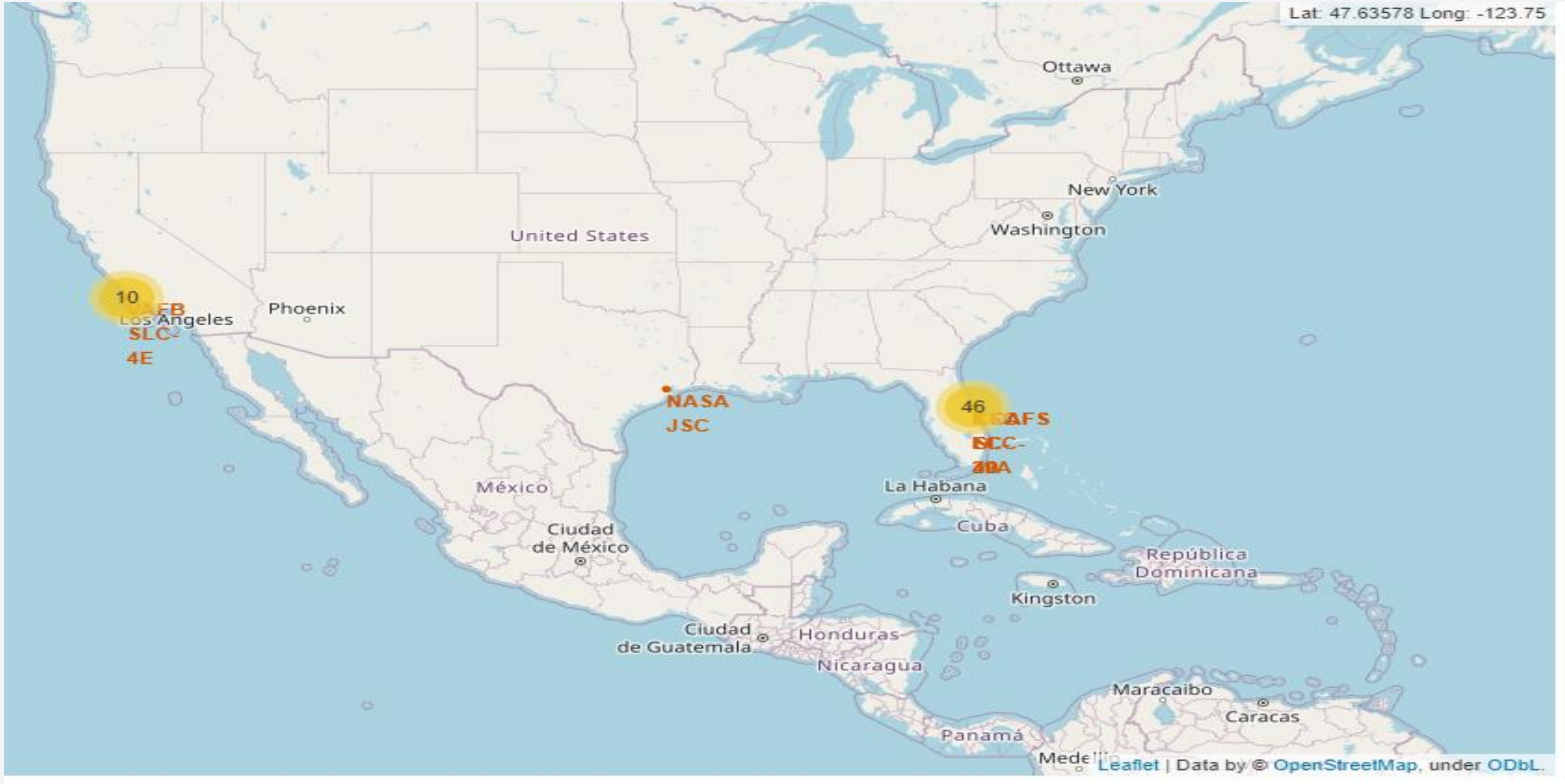
All launch sites on the map



The success/failed launches for each site on the map



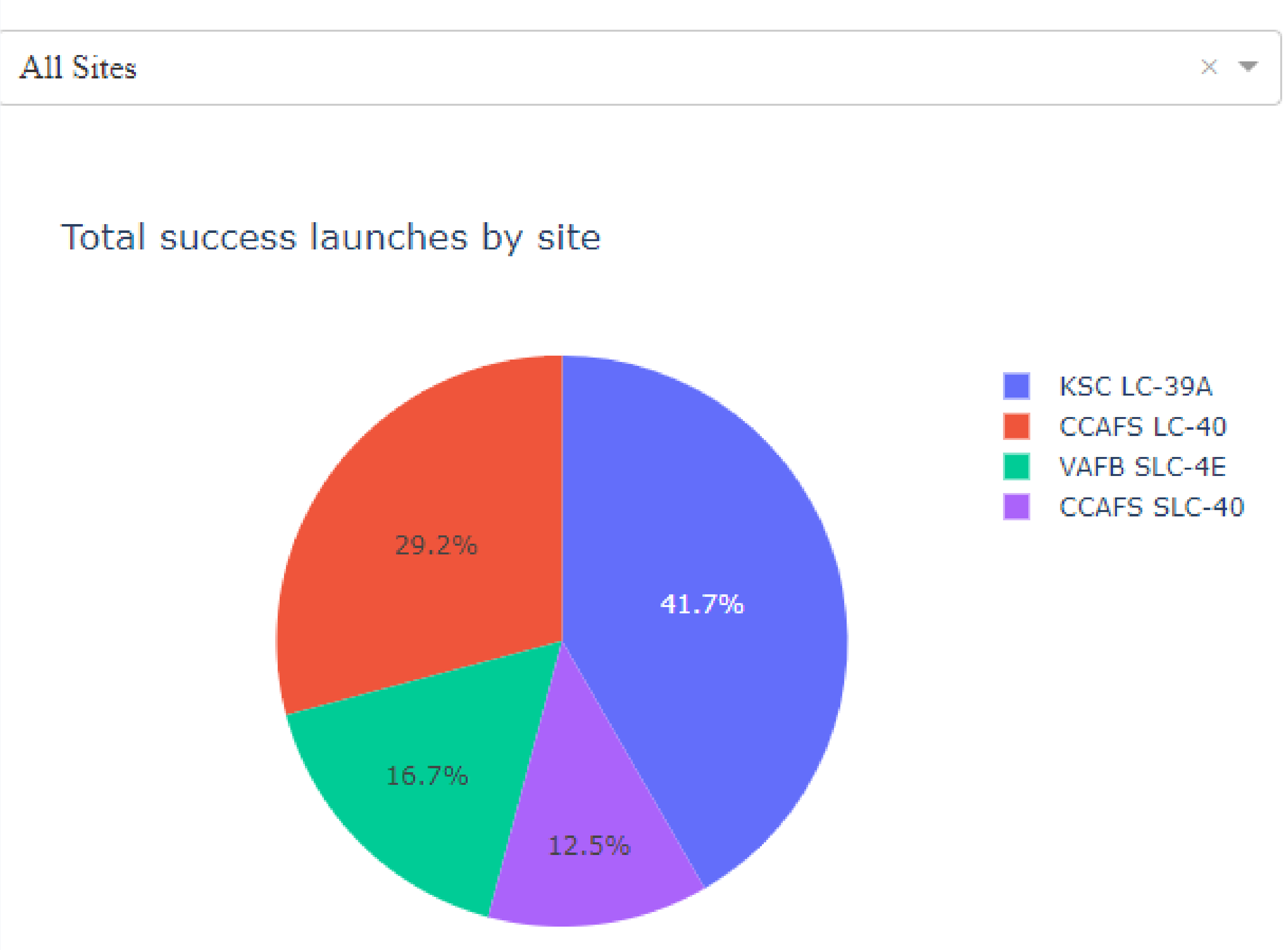
The distances between a launch site to its proximities



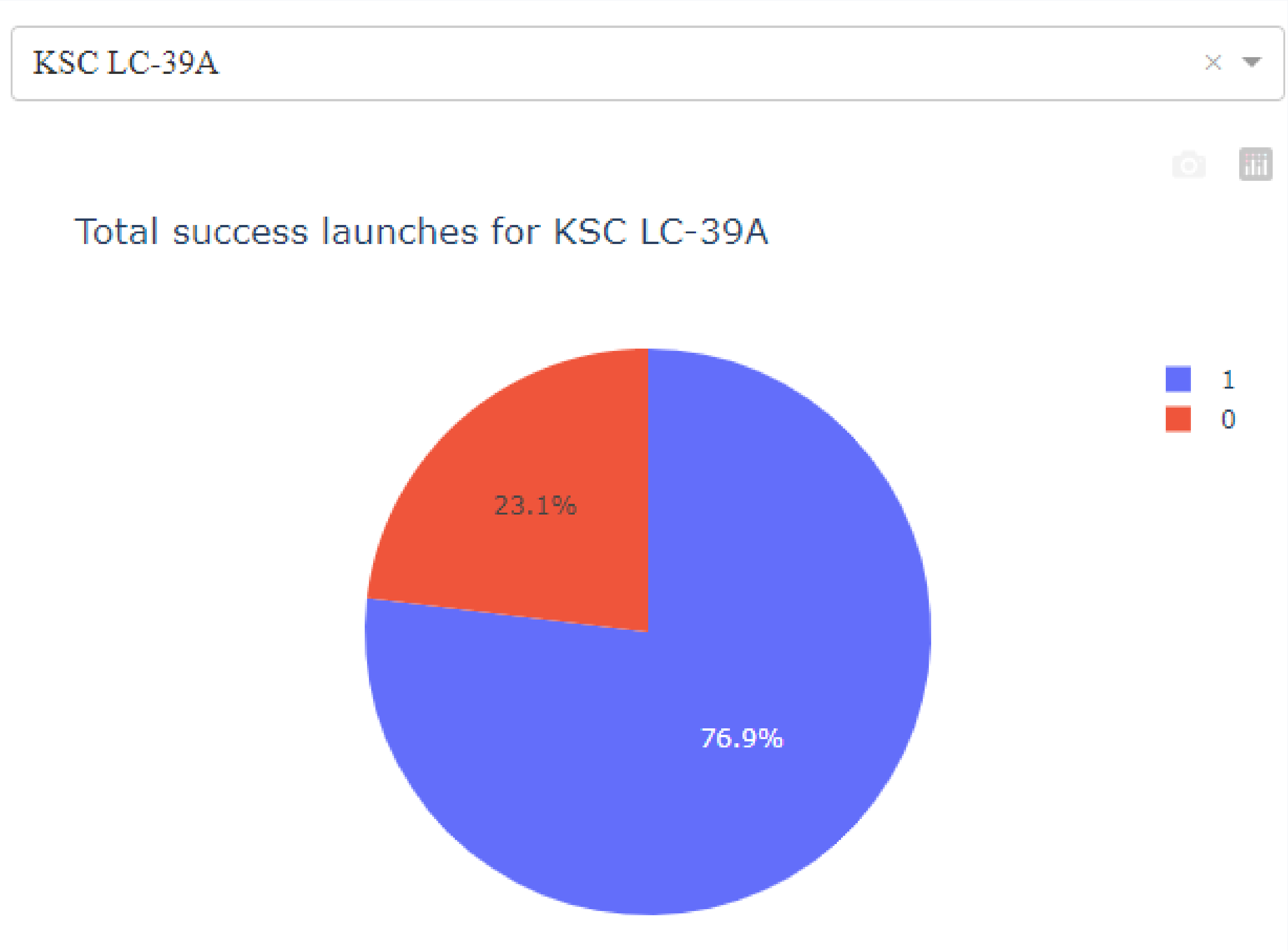
Section 4

Build a Dashboard with Plotly Dash

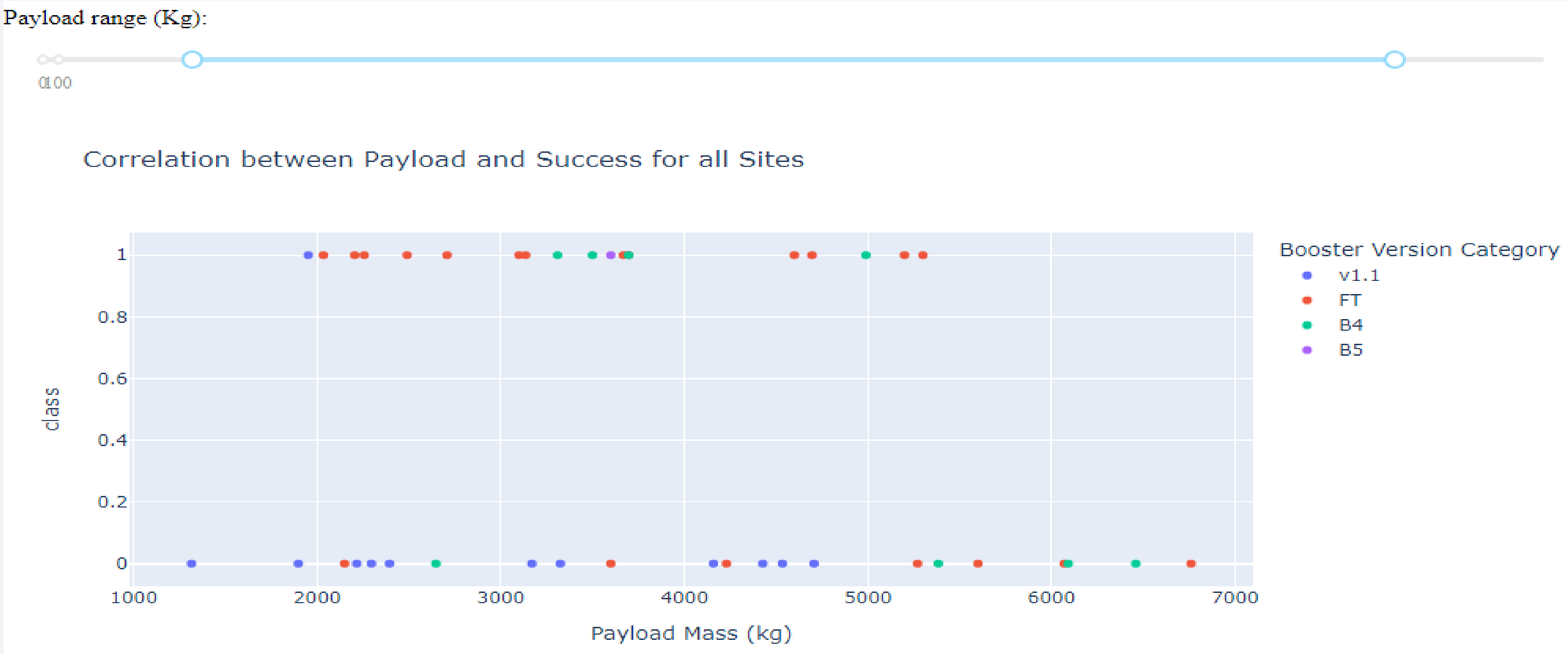
Launch success count for all sites



Launch site with highest launch success ratio



Payload vs. Launch Outcome scatter plot for all sites

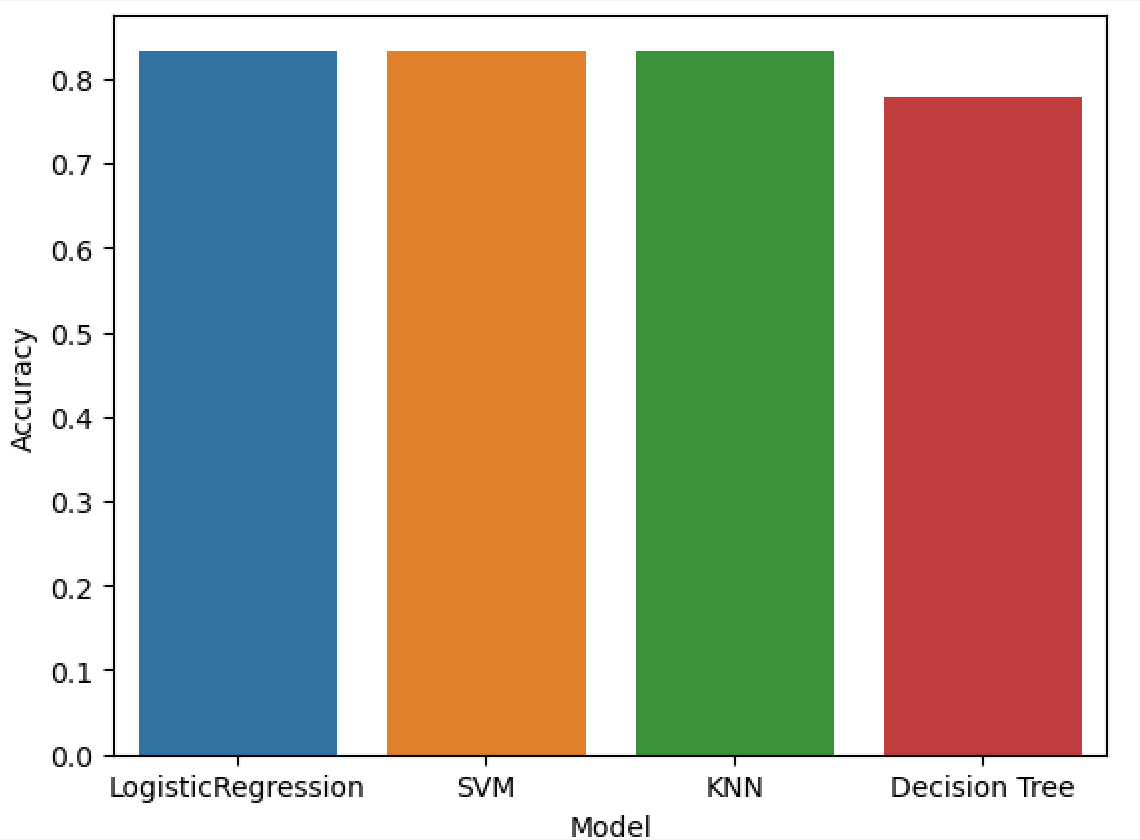


Section 5

Predictive Analysis (Classification)

Classification Accuracy

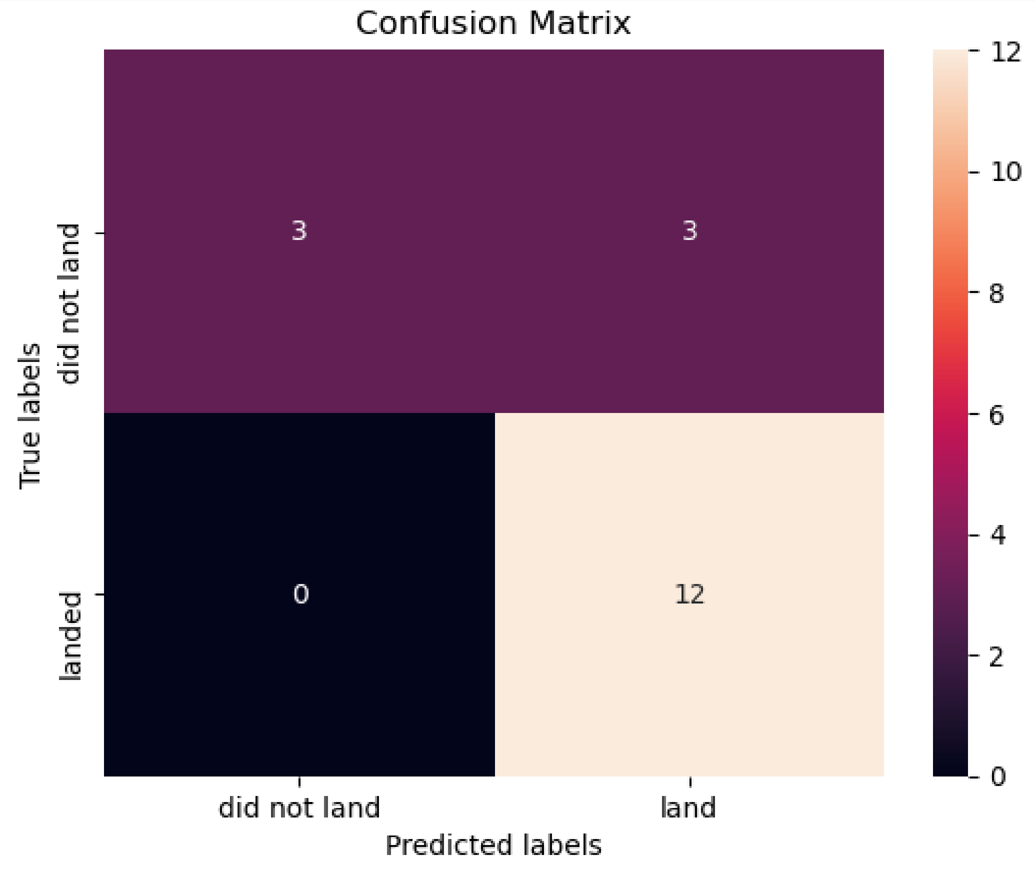
- The built model accuracy for all built classification models, in a bar chart



- All the models except Decision Tree performed well with an accuracy of 83%

Confusion Matrix

- The confusion matrix of the best performing model with an explanation



Conclusions

- SpaceX can rely on either Logistic Regression, Support Vector Machines or K-Nearest Neighbors classifiers to predict whether the first launch will be successful or not.
- However more model improvement techniques such as feature engineering, data wrangling, etc need to be performed to increase the accuracy of the models.

Appendix

- Relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets created during this project.

```python
# TASK 4:
# Add a callback function for `site-dropdown` and `payload-slider` as inputs, `success-payload-scatter-chart` as output
# Function decorator to specify function input and output
@app.callback(Output(component_id='success-payload-scatter-chart', component_property='figure'),
                [Input(component_id='site-dropdown', component_property='value'),
                Input(component_id="payload-slider", component_property="value")]
               )
def get_scatter_plot(entered_site, payload):
    # filtered_df = spacex_df
    filtered_df = spacex_df[(spacex_df['Payload Mass (kg)'] >= payload[0]) & (spacex_df['Payload Mass (kg)'] <= payload[1])]

    if entered_site == 'ALL':
        print(payload, type(payload))

        fig = px.scatter(filtered_df, x='Payload Mass (kg)', y='class', color='Booster Version Category',
        title='Correlation between Payload and Success for all Sites')
        return fig
    else:
        # return the outcomes scatter plot for a selected site
        print(payload, type(payload))
        filtered_df = filtered_df[spacex_df['Launch Site'] == entered_site]
        fig = px.scatter(filtered_df, x='Payload Mass (kg)', y='class', color='Booster Version Category',
        title=f'Correlation between Payload and Success for {entered_site}')
        return fig
```

Thank you!